



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/727,723	12/01/2000	Daniel A. Nagel	CIS00-3681	7519

7590

06/03/2005

Barry W. Chapin, Esq.  
Chapin & Huang, L.L.C.  
Westborough Office Park  
1700 West Park Drive  
Westborough, MA 01581

EXAMINER

CHOUDHURY, AZIZUL Q

ART UNIT

PAPER NUMBER

2145

DATE MAILED: 06/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/727,723

Applicant(s)

NAGEL ET AL

Examiner

Azizul Choudhury

Art Unit

2145

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 26 January 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 December 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

***Detailed Action***

This office action is in response to the amendment received on January 26, 2005.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-37 are rejected under 35 U.S.C. 102(b) as being anticipated by DuFresne (US Pat No: US005835712A), hereafter referred to as DuFresne.

1. With regards to claims 1, 10, 19 and 20, DuFresne teaches a method for transferring information between logic entities in browser pages, the method comprising the steps of defining a data element having a value for use by a first application logic entity in a first browser page; generating a browser page identifier for a second browser page, the browser page identifier including the value for the data element; invoking access to a second browser page using the browser page identifier, the second browser page including a second application logic entity; and retrieving the value of the data element from the browser page identifier for use by the second application logic entity.

(DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure

18, data is entered into the inputs of a first webpage. Also in the first webpage is the state 1, which is equivalent to the claimed browser page identifier. The input data is combined with state 1 and a script (equivalent to the claimed application logic entity) to create a new state, state 2. This state 2 is equivalent to the claimed second browser page identifier. As is seen in Figure 14, besides scripts, templates are also applicable).

2. With regards to claims 2 and 11, DuFresne teaches a method wherein the step of defining a data element includes the steps of: declaring data element for use by a first application logic entity; and providing a value for the data element (DuFresne discloses a design where data fields are present by which to enter and thus provide data elements (shown as inputs) (Figure 18, DuFresne). The script combines the inputs to create a new state (a new browser identifier)).

3. With regards to claims 3 and 12, DuFresne teaches a method wherein the step of providing a value for the data element comprises the step of: retrieving the value for the data element from a browser page identifier identifying the first browser page (DuFresne's design sends the new state to the second web page (Figure 18, DuFresne). This new state has the old state and the inputs (data elements) combined in it and are hence retrievable).

4. With regards to claims 4 and 13, DuFresne teaches a method wherein the step of retrieving a value of the data element from a browser page identifier identifying the first browser page includes the steps of: parsing the browser page identifier to retrieve a value for a data element from the browser page identifier; assigning the value to the data element that corresponds to the value parsed from the browser page identifier; and repeating the steps of parsing and assigning for each value contained in the browser page identifier such that all data elements containing a value within the browser page identifier receive an assignment of their respective value parsed from the browser page identifier (DuFresne's design uses scripts to combine the inputs (data elements) and the state (browser identifier) (Figure 18, DuFresne). The design also allows the scripts to retrieve the data values (Figure 14, DuFresne)).
5. With regards to claims 5 and 14, DuFresne teaches a method wherein the step of generating a browser page identifier includes the steps of: extracting a value for each data element shared between the first application logic entity and the second application logic entity to create a value array; obtaining a page designator for the second browser page; and appending the value array containing the values for each data element to the page designator for the second browser page to form the browser page identifier (DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure 18, data is entered into the inputs of a first

webpage. Also in the first webpage is the state 1, which is equivalent to the first browser page identifier. The input data is combined with state 1 and a script (equivalent to the application logic entity) to create a new state, state 2. This state 2 is equivalent to the second browser page identifier; it contains the old state and the inputs (values)).

6. With regards to claims 6 and 15, DuFresne teaches a method wherein the browser page identifier is a uniform resource locator that is dynamically generated via the steps of extracting, obtaining and appending and that contains the value of the data element shared by the first application logic entity and the second application logic entity (DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure 18, data is entered into the inputs of a first webpage. Also in the first webpage is the state 1, which is equivalent to the first browser page identifier. The input data is combined with state 1 and a script (equivalent to the application logic entity) to create a new state, state 2. This state 2 is equivalent to the second browser page identifier. DuFresne also discloses that state information is embedded into web pages for identification purposes (column 20, lines 4-7, DuFresne). Hence, just as URLs are page identifiers, in DuFresne's designs, states are page identifiers).

7. With regards to claims 7 and 16, DuFresne teaches a method further including the step of detecting a navigation command to navigate to the second browser page; and

wherein the steps of generating and invoking are performed in response to the step of detecting the navigation command to navigate to the second browser page, such that the browser page identifier produced in response to the step of detecting the navigation command includes a value for the data element that is created by the first application logic entity and is passed to the second application logic entity via the browser page identifier (DuFresne's design has a select button which invokes the process of creating a new state using the inputs and old state for a new web page (Figure 18, DuFresne)).

8. With regards to claims 8 and 17, DuFresne teaches a method wherein: the steps of generating and invoking are performed by a state sender logic entity; and wherein the step of retrieving is performed by a state retrieval logic entity; the state sender logic entity and state retrieval logic entity being logic entities incorporated into the first browser page and second browser page which interoperate to transfer values of data elements shared by the first application logic entity and the second application logic entity between the first browser page and the second browser page via incorporation of such values of data elements into browser page identifiers (DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure 18, data is entered into the inputs of a first webpage. Also in the first webpage is the state 1, which is equivalent to the first browser page identifier. The input data is combined with state 1 and a script (equivalent to the application logic entity) to create a new state, state 2.

This state 2 is equivalent to the second browser page identifier. As is seen in Figure 14, the scripts are able to retrieve the data values as well).

9. With regards to claims 9 and 18, DuFresne teaches a method wherein: the first application logic entity and the second application logic entity collectively form an application; and wherein the values of data elements shared by the first application logic entity and the second application logic entity collectively form state information that the state sender logic entity and the state retrieval logic entity can pass between the first browser page and second browser page via browser page identifiers for use by the application (The scripts of DuFresne's design create a new state that allows data to be passed from a first browser page to a second browser page (Figure 18, DuFresne)).

10. With regards to claims 21 and 27, DuFresne teaches the method wherein generating the browser page identifier includes generating the browser page identifier from within the first browser page in the browser, the method further comprising: displaying the browser page identifier in the first browser page displayed to a user; and receiving a selection by the user of the browser page identifier to retrieve the second browser page (In DuFresne's design, the first web page has a first state (first browser identifier) (Figure 18, DuFresne). When "select" is chosen by the user, the creation of the second state (second browser identifier) is performed (equivalent to the claimed selection by user)).



11. With regards to claims 22 and 28, DuFresne teaches the method further comprising passing the value of the data element from the first browser page of the browser to the second browser page of the browser by: appending the value of the data element associated with the first browser page to the browser page identifier; in response to receiving the selection by the user, utilizing the browser page identifier to retrieve the second browser page; and utilizing the data value in the second browser page of the browser (In DuFresne's design, the first web page has a first state (first browser identifier) (Figure 18, DuFresne). When "select" is chosen by the user, the creation of the second state (second browser identifier) is performed (equivalent to the claimed selection by user). The script also enables the retrieval of the data value).
12. With regards to claims 23 and 29, DuFresne teaches the method, wherein the steps of defining, generating, invoking, and retrieving support a step of: exchanging data from script code associated with the first browser page to the second browser page not currently being viewed by a user of the browser (In DuFresne's design, the data is transferred from a first page to a second page thanks to a script and the second page is not necessarily viewable by the user (Figure 18, DuFresne)).
13. With regards to claims 24 and 30, DuFresne teaches the method, wherein the steps of defining, generating, invoking, and retrieving support a step of: maintaining state

information between a portion of script code associated with the first browser page of the browser and another portion of script code associated with the second browser page of the browser (DuFresne's design features scripts that handle the inputs (data) and the state (browser identifier) (Figures 18, DuFresne)).

14. With regards to claims 25 and 31, DuFresne teaches a method further comprising: from a state sender logic entity of the first browser page associated with the browser, appending the value of the data element to the browser page identifier; and from a state retrieval logic entity of the second browser page associated with the browser, receiving the value of the data element along with the browser page identifier (DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure 18, data is entered into the inputs of a first webpage. Also in the first webpage is the state 1, which is equivalent to the first browser page identifier. The input data is combined with state 1 and a script (equivalent to the application logic entity) to create a new state, state 2. This state 2 is equivalent to the second browser page identifier. As is seen in Figure 14, the scripts are able to retrieve the data values as well).

15. With regards to claims 26 and 32, DuFresne teaches a method further comprising: from a state sender logic entity of the second browser page associated with the browser, appending the value of the data element to a browser page identifier

associated with a third browser page; and from a state retrieval logic entity of the third browser page associated with the browser, receiving the value of the data element along with the second browser page identifier (DuFresne's design is able to send data from a second web page to a third web page (Figure 17, DuFresne)).

16. With regards to claim 33, DuFresne teaches a computer system further supporting:

from a state sender logic entity of the first browser page associated with the browser, appending the value of the data element to the browser page identifier used to retrieve the second browser page; and from a state retrieval logic entity of the second browser page associated with the browser, receiving the value of the data element appended to the browser page identifier for the second browser page by parsing the browser page identifier and retrieving the data element (DuFresne teaches a design allowing for data to be transferred from one webpage to another (column 19, line 50 – column 20, line 37, DuFresne). As is seen in Figure 18, data is entered into the inputs of a first webpage. Also in the first webpage is the state 1, which is equivalent to the first browser page identifier. The input data is combined with state 1 and a script (equivalent to the application logic entity) to create a new state, state 2. This state 2 is equivalent to the second browser page identifier. As is seen in Figure 14, the scripts are able to retrieve the data values as well).

17. With regards to claim 34, DuFresne teaches a computer system further supporting:

from a state sender logic entity of the second browser page associated with the

browser, modifying the value of the data element, appending the modified value of the data element to a browser page identifier used to retrieve a third browser page of the browser; invoking access to the third browser page to the browser page identifier associated with the third browser page; and from a state retrieval logic entity of the third browser page of the browser, receiving the modified value of the data element appended to the browser page identifier from the third browser page of the browser (DuFresne's design is able to send data from a second web page to a third web page (Figure 17, DuFresne)).

18. With regards to claim 35, DuFresne teaches a computer wherein the modified value includes a concatenation of a first string of data associated with the first browser page and second string of data associated with the second browser page (The inputs are combined with the state by the script to create a new state for the second web page in DuFresne's design (Figure 18, DuFresne)).

19. With regards to claim 36, DuFresne teaches a computer wherein the modified value includes an arithmetic result based on use of an arithmetic operation on a first numerical value associated with the first browser page and a second numerical value associated with the second browser page (DuFresne's design allows for the use of scripts and scripts are able to perform a variety of tasks (column 16, lines 17-29, DuFresne)).

20. With regards to claim 37, DuFresne teaches a computer system wherein generating the browser page identifier for the second browser page includes providing a delimiter between a first portion and a second portion of the browser page identifier, the first portion of the browser page identifier used by a server to serve the second browser page to the browser, the second portion of the browser page identifier including the value of the data element being passed from the first browser page to the second browser page, the value of the data element being ignore by a server serving the second browser page to the browser (The inputs and state are combined by the script to create a new script (Figure 18, DuFresne). In addition, the script is able to retrieve data values (Figure 14, DuFresne), otherwise there is no point in transferring the data values. Since the data value (inputs) are combined with other data yet is still retrievable, it is inherent that a delimiter of some form is applied by the script to distinguish between the data types).

### ***Response to Remarks***

The remarks along with the amendments filed on January 26, 2005 have been carefully reviewed but are not deemed fully persuasive. The examiner appreciates the efforts made to further clarify the claimed invention within the remarks submitted. In response, the examiner reevaluated the DuFresne art and remains convinced that it is equivalent to the claimed invention. This is not to state that the DuFresne art is equivalent to the invention disclosed within the specifications. To clarify the examiner's

Art Unit: 2145

standpoint on the matter, the office action has been revised with the same prior art to better illustrate how the claimed invention lacks novelty against the DuFresne art.

In addition, the applicants further remark that the respective steps claimed are performed within the browsers and are performed by logic entities. After reevaluation of the DuFresne art, the examiner believes that traits claimed in such language are present within the prior art. The design disclosed by DuFresne makes use of scripts, which are logic entities, and it is well known in the art that scripts are able to exist within browsers. It is the scripts of DuFresne's design that perform the data combination and transfer from one web page to another.

As for the newly added claims, they too feature broad claim language and in general reiterate the same traits presented within the previous claims. Hence, they too lack novelty against the DuFresne prior art.

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

Art Unit: 2145

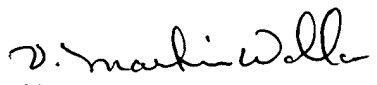
the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is (571) 272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Valencia Martin-Wallace can be reached on (571) 272-6159. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AC

  
VALENCIA MARTIN-WALLACE  
ADVISORY PATENT EXAMINER